Web Based Generic File Protection System

Bimali Arsakularatne, Kasun De Zoysa, Rasika Dayarathna

Department of Communication and Media Technologies,
University of Colombo School of Computing,
35, Reid Avenue, Colombo 7, Sri Lanka.

E-mail: wbimsdfa@yahoo.co.uk, kasun@cmb.ac.lk, rasika@cmb.ac.lk

Abstract

This document describes a web based generic file protection system, which uses simple cryptographic mechanisms to provide much more stronger security services.

Web technologies are used instead of a standalone approach since it does not require installing any additional software. In addition, a Web based approach allows the users to store critical information in a secure server instead of the local hard-disk. This not only protects critical data, but also enables the user to access these data from anywhere in the world.

At present, the following generic security services are provided for registered users:

- ***Integrity Checking Service**: used to detect unauthorized alteration of file systems.*
- ***Digital Signature Service**: useful for providing authorization and authenticity of documents.*
- ***Encryption Service**: provides protection against unauthorized access*.

Keywords: Digital Signatures, Encryption, Data Integrity, Document Protection

1. Introduction and Motivation

There are various types of software developed to ensure the security of a computer system. Among these are virus scanners, intrusion detection systems, firewalls etc. Even though a computer is secured using all or some of these software, can we make sure that it is adequately protected? For example, how do we know that hacker did not enter the system and altered the file-system without leaving any evidence?

Data in your computers can be compromised in the following ways.

- **Eavesdropping**: Information remains intact, but its privacy is being compromised. For example, someone could secretly read your personal emails without altering them.
- **Tampering**: Information is changed or replaced without your knowledge.
- **Impersonation**: Information passes to a person who poses as the intended recipient.

There are three cryptographic mechanisms, if used effectively, would ensure data protection in both storage and transmission of the data. They are described below.

Integrity Checking

The file system of a computer contains all of the long-lived data in the system including all user data, application data, system executables and databases. Therefore the file system is one of the usual targets of an attack. Motives for altering system files are many. Intruders could modify system databases and programs to allow future entry. System logs could be removed to cover their tracks or discourage future detection. Compromised security can lead to faulty services. Therefore the integrity of a file system should be closely monitored.

Integrity checking is most useful when tampering is so carefully and ingeniously carried out such that common means such as the 'last modified date' of a file or logs cannot identify any changes.

Digital Signatures

Although the use of digital signatures is not very popular in Sri Lanka and there is no legal enforcement based on digital signatures yet, we should anticipate and be prepared for the future. The value of digital signatures is immeasurable when it comes to e-business whether or not the business is web based. For example, from their experience of dealing with organizations of all sizes, it became apparent to DocumentFlow (a major software company in the

United States) that designers wanted to issue AutoCAD drawings to clients and contractors, but were reluctant to do so for fear of modifications to those drawings after approval [4].

Data Encryption

Data encryption plays a major role in e-business as well as within an organization in the process of passing messages between users. We often take email for granted and send confidential messages via email. But if we were to carry out the same process manually, we would take so many security precautions.

2. Background Theory

We start by introducing some concepts that are used throughout this paper.

2.1 Security Services

Data Integrity
A level of assurance that ensures information has not been deliberately or inadvertently modified or replaced in transit or storage.

Authentication
The process of proving one's identity.

Non-repudiation
The capability to demonstrate that an action such as the sending of a message was performed by a person with a particular identity.

The actual process of how these tasks are accomplished will be discussed consequently.

2.2 Secret Key Cryptography

In some encryption algorithms, the encryption key and the decryption key is the same, or the decryption key can be calculated from the encryption key. These algorithms are known as secret key algorithms (or private key algorithms/symmetric key algorithms). The encryption key must be kept secret and the sender and receiver must coordinate the use of their keys.

Following are some popular secret key algorithms [1]:

DES
This is the **D**ata **E**ncryption **S**tandard algorithm. There are known ways to attack this encryption. But they require a lot of computing power to do so.

DESede
This is also known as Triple-DES or multiple-DES. This algorithm uses multiple DES keys to perform three rounds of DES encryption or decryption. The added complexity increases the time required to break the encryption as well as the time required to encrypt or decrypt data.

PBEwithMD5andDES
This algorithm uses a password, a byte array known as a salt and an iteration count along with an MD5 message digest to produce a DES secret key. This key is used to perform DES encryption and decryption.

Blowfish
This algorithm is best used in applications where the key does not change often. It requires a lot of memory.

2.3 Public Key Cryptography

One of the most important breakthroughs in cryptography during the 20th century was the development of public key encryption. Public key algorithms (or asymmetric key algorithms) use separate keys for encryption and decryption. The encryption key is referred as the public key and the decryption key is called the private key. That is because when data is encrypted using the encryption key, you need the decryption key for the decryption of the data. Hence one can make his/her encryption key known to the world at large and therefore it is called the public key. On the other hand, one should protect his/her private key because data can be decrypted with it. The important feature in public key cryptography is that the sender and receiver do not have to share keys.

Figure 1 shows how public key cryptography works.

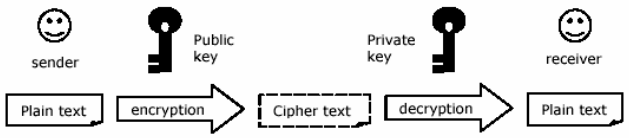

Figure 1: Operation of public key cryptography

RSA (Rivest, Shamir, Adleman)

The most famous public key algorithm – developed in 1977. The following is a summary of how RSA works [2]:

- Two large (100 digits or more) prime numbers p and q are generated with
  $n = pq$
- A public key e is selected as an integer such that e is relatively prime to $(p-1)(q-1)$
- The private key d is computed such that $ed \bmod ((p-1)(q-1))$ is 1
- Encryption is performed on plaintext numbers m that are smaller than n by calculating $m^e \bmod n$
- Decryption is performed on cipher-text c by calculating $c^d \bmod n$

Disadvantages of RSA

- RSA is 100 to 1000 times slower compared to secret key algorithms [2]
- Although RSA has not been broken, if an efficient way to factor large numbers was discovered, it could be easily broken.

2.4 Message Digests

Message digests are used to secure data integrity. In other words, to detect whether data has been modified or replaced.

A message digest is a special kind of function referred to as a one-way (hash) function. A one-way function is easy to calculate, but difficult to reverse. Message digests take messages or data as inputs and compute values referred to as hash values that are used as fingerprints to the messages.

Good message digests have the following properties.

- Given a particular hash value, it is computationally infeasible to compute the message that produced that value
- It is computationally infeasible to find two messages that yield the same hash value.

Examples of Message Digest Algorithms

- SHA1
- MD5

2.5 Secure Message Digests

A secure message digest is called a Message Authentication Code (MAC). A MAC has the property that it cannot be created solely from the input data. It requires a secret key that is shared by the sender and receiver. Hence an intermediate party cannot change both the data and the MAC without the receiver detecting that the data has been corrupted.

2.6 Digital Signatures

Digital signatures are mainly used to prove to you that a message sent to you is created by a particular individual or an organization. If the receiver can verify the digitally signed message from the sender, then the receiver can make sure that the contents of the message are correct and authentic.

Digital signatures have the following properties similar to real world signatures [2].

- Unforgeability – because the signer uses his private key to sign, only he can sign with that key
- Verifiability – because the signer's public key is openly available, anyone with access to the message and signature can verify that the message was signed by the signer and that neither the message nor the signature has been altered.
- Single use – A signature is unique to a particular message.
- Non repudiation – After a signer has signed a message and the message and signature have been sent to others, the signer cannot claim that he did not sign the message.
- Sealing – A signed message is digitally sealed. It cannot be altered without invalidating the signature.

2.7 Digital Certificates

When you use public keys of others to encrypt data or verify their digitally signed documents, there is no way to make sure that the public key belongs to the particular person that you are referring to. This is where digital certificates come in. Digital certificates are messages signed by a Certification Authority (an entity that is trusted to verify that other entities are who they claim to be) that certify the value of an entity's public key.

Figure 2 illustrates the use of digital signatures.

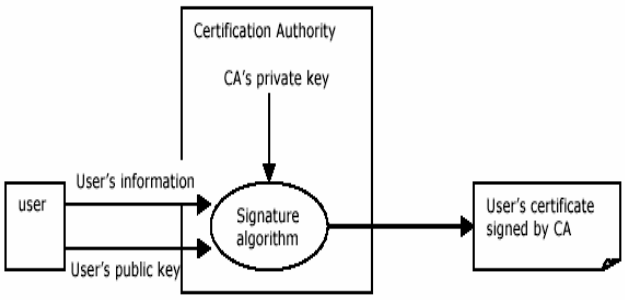

Figure 2: Use of digital signatures

2.8 Signed Applets

By signing an applet an organization or an individual can indicate that the organization has reviewed the applet for security and believes that the signed applet is free from security defects. The signature also implies that the organization/individual takes responsibility for the applet in cases where there is a security malfunction. Signing also provides the user with a mechanism for verifying that a signed applet originates from a particular organization and has been delivered to the user without modification. For these reasons, a user can determine that he or she is able to extend a certain level of trust to an applet that is signed by a reputable organization.

An applet is signed in two steps. In the first step, the applet's class files are archived in a JAR file. In the second step the JAR file is signed with a signing tool.

3. Literature Review

3.1 Review on Integrity Checking

Integrity analysis focuses on whether some aspect of a file or object has been altered. This often includes file and directory attributes, content and data streams. Integrity analysis often utilises strong cryptography mechanisms called **message digest** (or hash) algorithms that can recognize even subtle changes.

In simplest terms, a database is created with some unique identifier for each file to be monitored. By recreating that identifier (which could be a copy of the entire file contents) and comparing it against the saved version, it is possible to determine if a file has been altered or not. Furthermore, by comparing the entries in the database it is possible to determine if files have been added to or deleted from the system.

The file contents themselves are not usually saved, as this would require too much disk space. Instead, a checklist can be used. The checklist would contain a set of values generated from the original file (usually including the length, the time of last modification, owner etc.). It should be periodically regenerated and compared with the saved version and note what the differences are. However, changes may be made to the contents of files without any of these values (the checklist values) changing from the stored values [3]. Therefore to efficiently detect changes in the file system, values should be calculated from the contents of the files itself. If this value depends on the entire contents of the file and is difficult to match for an arbitrary change to the file, then storing this value is sufficient. This fingerprint or signature of the file can be saved instead of the file itself.

- The signature files used should be computationally simple to perform but infeasible to reverse.
- It should signal if the file changes
- It should be sufficiently large as to make chance collision unlikely.

3.2 Issues faced by an Integrity Checking Software

Database Issues

If the database can be updated after every authorized file addition, update or deletion, this prevents the change being reported in future checking events and hence reduces the complexity of the report issued.

But the database should also be secured from tampering. If a copy of the database can be kept remotely or on a removable disk, this problem can be solved.

Signature spoofing

Intruders could modify a file and remain undetected in an integrity checking scheme using file signatures if the file can be further modified to generate the same signature as the original. Two methods for finding such a modification are:

- Brute force search
- Inverting the signature function and spoofing the signature function

For these reasons, message digest algorithms become valuable as integrity checking tools. Message digests are usually large, often at least 128 bits in length and computationally infeasible to reverse and carry out a search.

Duplicate Search

The large number of collisions for the 16 bit signatures and the absence of any collisions for the 128 bit signatures confirm the expected observation that larger signatures are less likely to collide by accident.

Other Issues

The Integrity checker should only report, and not effect changes. Although a user could use the tool's output to drive changes, the tool itself would not provide any explicit means of making alterations to the system.

3.3 Review on Digital Signatures

There are many types of software developed so far to support the use of digital signatures. But a major flaw in them is that they do not support signing all types of

files. Some software supports digital signing of specific file types while others supports signing of document types only.

For example, *Adobe Digital Signature Architecture* provides a plug-in to sign PDF files [5]. *CADSign* enables AutoCAD file signing [4]. *Approve Desktop* claim to sign any type of document such as Word, Excel, PDF, AutoCAD, XML, HTML etc.

A reliable digital signature scheme should be able to sign any type of file whether it is a document type or an executable although it is true that documents are the most frequently used file type that needs digital signing when businesses and organizations are concerned.

Since a file can be signed by many signers, we should be able get information about each of the signatures on a file. This information should include the identification of the signer, signature date and time etc.

3.4 Review on Data Encryption

There are many types of software that support data encryption. But since it is one of the most essential components of a good file protection system, encryption facilities will also be included in the system being developed.

Some encryption algorithms can be broken in a matter of hours; some would take many years. Others would take several times as the anticipated lifetime of the universe to break, given machines many times more powerful than the ones in use today. Of course, the price you pay for more security is the encryption time, among other things. If the data will be useless in an hour, you do not need an algorithm to protect it for your lifetime.

Some algorithms are prohibitively slow for common use. If you need a Cray mainframe to encrypt and decrypt the data in a reasonable time, it probably is not a good choice for an applet.

4. The System Architecture

The system is web-based so that everybody can enjoy the security services provided. A web site is developed which supplies the following services to its users.

- Obtaining membership with the server
- Generating symmetric and asymmetric keys
- Digitally signing documents and uploading them to the server
- Verifying digitally signed documents
- Providing encryption and decryption facilities
- Scanning for integrity of specified files
- Updating the integrity status of selected files
- Calculate hash value of specified files

4.1 Why web based?

By allowing the users to download the programs as signed applets, it can be guaranteed that they have access to the latest version of the software and that it is not altered by some third party. Another important advantage of a web based system is that users can access the system from anywhere in the world to use the provided services.

4.2 Remote Database Storage

Users may find their computers as an insecure place to store critical data items such as keys, digital certificates etc. Our system provides a facility to store their critical data items at a remote database if they prefer to do so. The remote database can be located at the same machine as the web server or at some other location.

It should be pointed out that a remote database is vital for the integrity checking operation. The reason for this is that if the message digests were stored locally, a hacker who changes a file can change the corresponding message digest so that a change would not be reflected. Therefore the message digests should be stored more securely.

4.3 Client Side

Applets are used in the client end of the system. A bulk of the processing is being done in the applets. This helps to reduce processing weight on the server as well as unnecessary network traffic since most operations do not need to access the server. For example, encrypting a file can be processed totally at the client side if the necessary keys are located at the client. Otherwise, if the keys are located at the server, the applet just has to download the necessary key from the server and carry out the operation.

Since ordinary applets restrict many vital operations (such as file accessing at the client side) signed applets are used. All the operations that can be performed by a normal java application can also be performed using signed applets.

4.4 Server Side

The server side will be manipulated via Java Servlets. Since Servlets provide multi-threaded access by default this was a natural choice. The servlets have to handle database access depending on the call from the clients.

4.5 Data Transfer

The applets generate a symmetric key at initialization. Then this key is encrypted with the server's public key and is sent to the server. Thereafter the applet and the server communicate with each other using the symmetric key to encrypt data (see Figure 3).

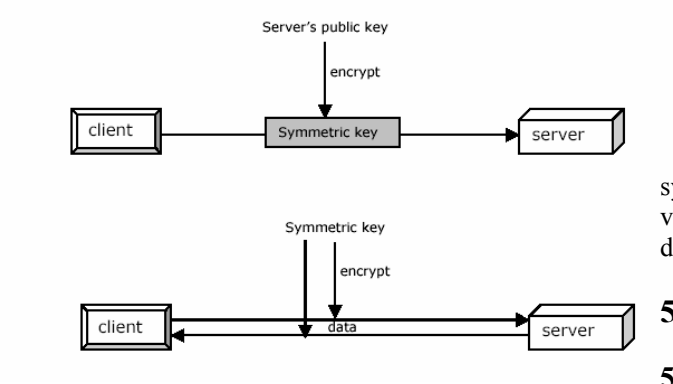

Figure 3: Data transmission protocol

4.6 Integrity Checking Process

- The integrity checking is handled using message digests.
- Initially, the user has to store the state of his file-system when the file-system is at a safe state (i.e. the user is certain that the file system has not been changed without his/her knowledge). At this stage, a message digest for each file is calculated and stored at the server database (see Figure 4).

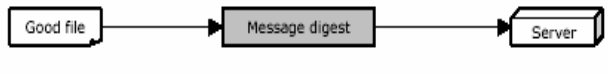

Figure 4: Update of file status

- When the file system is to be scanned for integrity checking, a set of message digests will be calculated for each file once more. Then these message digests will be compared with the ones stored at the remote server. If these two are different for a certain file, then the corresponding file has been changed (See figure 5).

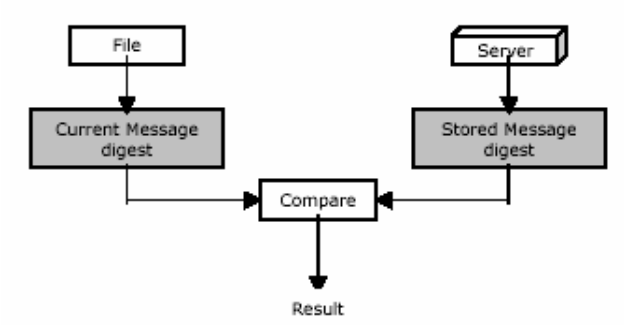

Figure 5: Integrity checking

In addition to the integrity checking process, the system also supports document signing and verification as well as documents encryption and decryption.

5. Functionality of the System

5.1 Obtaining Membership with the Server

Every user of the system has to register with the system and obtain a unique username. To register, users must provide personal information such as full name, address, nationality, occupation etc. Users must provide a password which will be used in subsequent login sessions.

5.2 Generating Symmetric and Asymmetric Keys

User selects the type of key that should be created. E.g. Symmetric/asymmetric. Next, the key algorithm to be used and the location where the key should be stored has to be given. Finally, the keys are generated. Key name is validated and checked for duplications.

5.3 Signing and Verifying Documents

User selects the file to be signed and the private key to sign with. This private key can be one which is stored locally or the user can download one of his/her pre-generated keys from the server. After that the user has to choose to either save the signed file or to upload it to the server. The system will act accordingly.

In the case of verification, user selects the signed file and the relevant public key or certificate to verify with. After the verification, the system indicates to the user whether the verification was successful or not.

5.4 Encryption and Decryption Facilities

User selects the file to encrypt. Then chooses the key type (symmetric or asymmetric) and the key algorithm

(DES, DESede etc. or RSA). Then the applet will save the encrypted file as specified by the user.

In decryption, user selects the encrypted file and the relevant key. The system will decrypt the file and save it on the hard disk.

5.5 Scanning for Integrity of Specified Files

User selects the files that should be scanned. He is given 4 options to:

- Scan the complete file system
- Scan critical files
- Scan selected partitions
- Scan selected files and directories

The results will be displayed in a report. User can update the status of the scanned files by clicking a button on the report.

6. Featured Attributes of the System

The special features of this system that make it outstanding from the rest are discussed below and the above-mentioned qualities will be addressed as appropriate.

6.1 Remote Storage of Critical Data

Many computer users have to share the same computer with several people – may it be at the office, college or at home. In such situations, they face the problem of not being able to protect their important files. Some machines, even if used by one person, if it is on a network, could be open to trespassing of others. Suppose they choose to hide their data using some kind of key – based encryption method, still their keys would be in danger. Likewise, if the hash values that represent the file system integrity are stored in such an unsafe computer, there would be no use of integrity checking either.

Therefore it can be seen that it would be very convenient if there was a safe place that we can store our critical data items and retrieve them whenever we want to. The developed system provides this facility of remote storage of critical data. Of course, the system administrator has to ensure that the machine with the database is fully protected and the users have to be confident that their data are safe at the server.

6.2 Platform Independence

Since this system is developed using Java applets, it is platform independent and can be executed in any operating system including Unix, Linux, Windows platforms etc. This is a very important issue because most networks contain nodes with different operating systems and if the security services cannot be utilized in one of them, then the whole network could be compromised.

6.3 Use of Key - Based Encryption

Many software that provide data encryption facilities operate with the use of passwords. But there are freely available software that can be used to crack a particular password. Key based algorithms are more powerful because it is not possible to attack key based algorithms easily. Even though there are known ways of attacks for key types such as DES, such attacks need very powerful computers which are not available for the ordinary user.

6.4 Installation is not necessary

No software installation is necessary for the operation of the system. This is very convenient in contrast with the software that have to be installed in each and every computer. Another point to note is that by the use of signed applets, the software itself is protected from unauthorized alterations and that the users can ensure this by the use of digital signature verification.

6.5 Can sign or encrypt any type of file

Most currently available software can only sign several types of files, specially document types. (See section 3.2). But the developed system can sign or encrypt any type of file.

6.6 Ease of Integration

There are no complexities in system integration. The server contains the servlets that can be executed using any servlet enabled web server. After the initial installation of the server, all the other users can access the system through the web via their usual web browser. The important thing to note is that the JDK 1.4 plug-in is a necessity for this system to function properly as the earlier versions of Java do not contain the cryptographic extensions.

6.7 User-friendliness

The system is user-friendly and is usable by the average person. The good report structure provided in Integrity scanning is a good example for this and it is comprehensible by the average person. Tool-tips and help messages are given at necessary points and an online user manual is also available.

In general the system provides many facilities together with the ones which were mentioned in this

section and users just have to log on to the website to exploit them.

7. Conclusion

At present, Computer file protection is a very important requirement in all sorts of computer systems. The Web Based Generic File Protection System provides security services required for protection of computer files. The developed system is only a first attempt and it could be improved to a professional standard together with the enhancements discussed in section 8. Once improved, it can be used either in the Internet or within an intranet by an organization to protect their data and to carry out their day-to-day functions more confidently.

8. Future Enhancements

At the current stage, the system provides many services, but it can be enhanced making further improvements as discussed below.

- As stated in section 2.7, digital certificates provide a way of ensuring that a particular public key belongs to a specified party. The file protection system can be improved to support signature verification by digital certificates as well as public keys.
- Currently the system can scan only the machine by which the applet is being accessed. When it comes to scanning a large network, scanning machines one by one becomes a tedious task. The system can be improved with the use of mobile agent technology, such that a complete network could be scanned with one attempt.
- The system would be more useful for system administrators if there is a facility where one can enter the IP address or the location of a machine located remotely and then supervise the scanning remotely.
- Only few encryption algorithms are implemented in the system. It can be extended to support more.

References

1. "Java Security", 2nd edition, Scott Oaks, O'Reilly & Associates, Inc. May 2001
2. "Java Security Handbook ", Jamie Jaworski, Venkata S. Chaganti, Paul J. PerroneJamie, Macmillan, USA, September 2000
3. Design and Implementation of Tripwire: A file system integrity checker – Gene H. Kim and H. Spafford, Department of Computer Sciences, Perdue University
4. "Signed Off", Article from CAD User Magazine (www.cadserver.co.uk)
5. "Acrobat Digital Signature Overview", Technical Note #5400 (Adobe Systems Incorporated www.adobe.com)